

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A system ~~for processing data received in a plurality of incoming streams of variable speeds~~, comprising:
 - a memory configured to store data associated with a plurality of incoming streams of ~~variable~~ different speeds;
 - an interface controller comprising a first arbitration element to arbitrate among the streams ~~of variable speeds~~ to store the data in the memory, the first arbitration element including a number of first entries, one of the first entries indicating which of the streams is to be serviced in a particular first time slot, the streams being assigned to the first entries based on the speeds of the streams; and
 - a dispatch unit comprising a second arbitration element to arbitrate among the streams ~~of variable speeds~~ to read the data from the memory, the second arbitration element including a number of second entries, one of the second entries indicating which of the streams is to be serviced in a particular second time slot, the streams being assigned to the second entries based on the speeds of the streams.
2. (original) The system of claim 1, wherein the memory includes:
 - a plurality of memory buckets corresponding to the streams.
3. (original) The system of claim 2, wherein the memory buckets have a fixed size.

4. (currently amended) The system of claim 1, ~~wherein the first arbitration element is configured to store a plurality of entries~~, each of the first entries including a stream number that identifies one of the streams.

5. (currently amended) The system of claim ~~[[4]]~~ 1, wherein the number of the first entries in the first arbitration element ~~for a particular one of the streams is based on a speed of the stream~~ is programmable.

6. (original) The system of claim 4, wherein the interface controller is configured to:

read one of the stream numbers from the first arbitration element,
gather data corresponding to the identified stream, and
transfer the data to the memory.

7. (original) The system of claim 6, wherein the interface controller is further configured to send a stream identifier with the data transferred to the memory.

8. (original) The system of claim 7, wherein the memory is further configured to sort the data from the interface controller based on the stream identifier.

9. (original) The system of claim 1, wherein the first and second arbitration elements are synchronized.

10. (currently amended) The system of claim 1, ~~wherein the second arbitration element is configured to store a plurality of entries~~, each of the second entries including a stream number that identifies one of the streams.

11. (currently amended) The system of claim ~~[[10]]~~ 1, wherein the number of the second entries in the second arbitration element ~~for a particular one of the streams is based on a speed of the stream~~ is programmable.

12. (original) The system of claim 10, wherein the dispatch unit is configured to:
read one of the stream numbers from the second arbitration element,
read data corresponding to the identified stream from the memory, and
output the data for processing.

13. (original) The system of claim 1, further comprising:
flow control logic configured to initiate flow control on the storing of data in the memory.

14. (original) The system of claim 13, wherein the flow control includes dropping data from the stream.

15. (original) The system of claim 13, wherein the flow control includes causing the interface controller to stop storing data from the stream in the memory.

16. (original) The system of claim 13, wherein the flow control logic includes:
a buffer configured to temporarily store the data from the interface controller in a plurality of entries,
a counter configured to determine a number of entries in the buffer corresponding to each of the streams, and
a comparator configured to determine whether to initiate the flow control for each of the streams based on the determined number of entries for the stream.

17. (original) The system of claim 16, wherein the comparator is configured to compare the determined number of entries for a stream to a watermark and initiate the flow control for the stream when the determined number of entries exceeds the watermark.

18. (original) The system of claim 17, wherein the comparator is further configured to compare the determined number of entries for the stream to a second watermark and drop data from the stream when the determined number of entries exceeds the second watermark.

19. (currently amended) The system of claim 1, wherein each of the streams has an associated watermark for ~~use in~~ performing flow control on the storing of data in the memory.

20. (currently amended) The system of claim 1, wherein each of the streams has two associated watermarks for ~~use in~~ performing flow control on the storing of data in the memory.

21. (currently amended) A method ~~for processing data received in a plurality of incoming streams of variable speeds~~, comprising:

storing data from a ~~plurality of variable speed~~ a plurality of streams of potentially different speeds in a memory using a first arbitration ~~element that arbitrates among the variable speed~~ scheme that stores data associated with a faster one of the streams in the memory at a higher rate than data associated with a slower one of the streams; and

reading the data from the memory using a second arbitration ~~element that arbitrates among the variable speed~~ scheme that reads the data associated with the faster one of the streams from the memory at a higher rate than the data associated with the slower one of the streams.

22. (original) The method of claim 21, wherein the storing includes:
storing the data in a plurality of memory buckets in the memory based on the streams to which the data belongs.

23. (currently amended) The method of claim 21, further comprising:
writing a plurality of entries into ~~[[the]]~~ a first arbitration element that implements the first arbitration scheme, wherein each of the entries including a stream number that identifies one of the streams.

24. (currently amended) The method of claim 23, wherein the number of the entries in the first arbitration element for ~~a particular~~ the faster one of the streams ~~is based on a speed of the stream~~ being greater than the number of the entries in the first arbitration element for the slower one of the streams.

25. (original) The method of claim 23, wherein the storing includes:
reading one of the stream numbers from the first arbitration element,
gathering data corresponding to the identified stream, and
transferring the data to the memory.

26. (original) The method of claim 25, wherein the transferring includes:
sending a stream identifier with the data transferred to the memory.

27. (original) The method of claim 26, wherein the storing further includes:
sorting the data based on the stream identifier.

28. (original) The method of claim 21, wherein the first and second arbitration elements are synchronized.

29. (currently amended) The method of claim 21, further comprising:
writing a plurality of entries into ~~[[the]]~~ a second arbitration element that implements the second arbitration scheme, each of the entries including a stream number that identifies one of the streams.

30. (currently amended) The method of claim 29, wherein the number of the entries in the second arbitration element for ~~a particular~~ the faster one of the streams ~~is based on a speed of the stream~~ being greater than the number of the entries in the second arbitration element for the slower one of the streams.

31. (original) The method of claim 29, wherein the reading includes:
obtaining one of the stream numbers from the second arbitration element,
obtaining data corresponding to the identified stream from the memory, and
outputting the data for processing.

32. (original) The method of claim 21, further comprising:
initiating flow control on the storing of data in the memory.

33. (original) The method of claim 32, wherein the initiating includes:

temporarily storing the data in a plurality of entries in a buffer,
determining a number of entries in the buffer corresponding to each of the streams, and
determining whether to initiate the flow control for each of the streams based on the
determined number of entries for the stream.

34. (original) The method of claim 33, wherein the determining whether to initiate
the flow control includes:

comparing the determined number of entries for a stream to a watermark, and
initiating the flow control for the stream when the determined number of entries exceeds
the watermark.

35. (original) The method of claim 34, wherein the determining whether to initiate
the flow control includes:

comparing the determined number of entries for the stream to a second watermark, and
dropping data from the stream when the determined number of entries exceeds the second
watermark.

36. (original) The method of claim 32, wherein the initiating the flow control
includes:

dropping data from the stream.

37. (original) The method of claim 32, wherein the initiating the flow control includes:

stopping the storing of data from the stream in the memory.

38. (currently amended) The method of claim 21, wherein each of the streams has an associated watermark for ~~use in~~ performing flow control for the associated stream.

39. (currently amended) The method of claim 21, wherein each of the streams has two associated watermarks for ~~use in~~ performing flow control for the associated stream.

40. (original) A system for performing flow control on data in a plurality of incoming streams of variable speeds, comprising:

a buffer configured to temporarily store data from a plurality of streams of variable speeds in a plurality of entries;

a counter configured to determine a number of entries in the buffer corresponding to each of the streams; and

a comparator configured to determine whether to initiate flow control for each of the streams based on the determined number of entries for the stream.

41. (original) The system of claim 40, wherein the comparator is configured to compare the determined number of entries for a stream to a watermark and initiate the flow control for the stream when the determined number of entries exceeds the watermark.

42. (original) The system of claim 41, wherein the flow control includes dropping data from the stream.

43. (original) The system of claim 41, wherein the flow control includes causing a sender of the data to stop outputting data in the stream.

44. (original) The system of claim 41, wherein the comparator is further configured to compare the determined number of entries for the stream to a second watermark and drop data from the stream when the determined number of entries exceeds the second watermark.

45. (original) The system of claim 40, wherein each of the streams has an associated watermark for use in performing flow control on the associated stream.

46. (original) The system of claim 40, wherein each of the streams has two associated watermarks for use in performing flow control on the associated stream.

47. (original) The system of claim 40, further comprising:
a memory configured to store data from the buffer;
an interface controller having a first arbitration element and configured to output the data to the buffer using the first arbitration element; and

a dispatch unit having a second arbitration element and configured to read the data from the memory using the second arbitration element.

48. (original) A method for performing flow control on data in a plurality of incoming streams of variable speeds, comprising:

storing data from a plurality of streams of variable speeds in a plurality of entries of a buffer;

determining a number of entries in the buffer corresponding to each of the streams; and

determining whether to initiate the flow control for each of the streams based on the determined number of entries for the stream.

49. (original) The method of claim 48, wherein the determining whether to initiate the flow control includes:

comparing the determined number of entries for a stream to a watermark, and

initiating the flow control for the stream when the determined number of entries exceeds the watermark.

50. (original) The method of claim 49, wherein the initiating the flow control includes:

dropping data from the stream.

51. (original) The method of claim 49, wherein the initiating the flow control includes:

causing a sender of the data to stop outputting data in the stream.

52. (original) The method of claim 49, wherein the determining whether to initiate the flow control includes:

comparing the determined number of entries for the stream to a second watermark, and dropping data from the stream when the determined number of entries exceeds the second watermark.

53. (original) The method of claim 49, wherein each of the streams has an associated watermark.

54. (original) The method of claim 49, wherein each of the streams has two associated watermarks.

55. (original) A system for performing flow control on data in a plurality of incoming streams of variable speeds, comprising:

a buffer configured to temporarily store data from a plurality of streams of variable speeds in a plurality of entries;

a counter configured to determine a number of entries in the buffer corresponding to each of the streams; and

a comparator configured to:

- compare the determined number of entries for a stream to first and second watermarks,
- initiate flow control for the stream when the determined number of entries exceeds the first watermark, and
- drop data from the stream when the determined number of entries exceeds the second watermark.

56. (currently amended) A network device, comprising:

an input interface configured to:

- receive a plurality of packets belonging to a plurality of streams of ~~variable speed~~ differing speeds,

- access a first arbitration scheme that services a faster one of the streams more often than a slower one of the streams, and

- output the packets based on ~~[[a]] the first arbitration scheme that arbitrates among the streams of variable speed;~~

input logic comprising:

- flow control logic configured to initiate flow control on the packets output by the input interface,

- a memory configured to store the packets from the input interface, and

- a dispatch unit configured to;

access a second arbitration scheme that services the faster one of the streams more often than the slower one of the streams, and

read the packets from the memory based on ~~[[a]]~~ the second arbitration scheme ~~that arbitrates among the streams of variable speed;~~ and one or more packet processors configured to process the packets from the dispatch unit.

57. (currently amended) A network device, comprising:
- means for receiving a plurality of packets belonging to a plurality of streams of ~~variable speed~~ potentially different speeds;
 - means for storing the packets based on a first arbitration scheme that ~~arbitrates among the streams of variable speed~~ stores the packets based on the speeds of the streams to which the packets belong;
 - means for performing flow control on the storing of the packets;
 - means for reading the packets based on a second arbitration scheme that ~~arbitrates among the streams of variable speed~~ reads the packets based on the speeds of the streams to which the packets belong; and
 - means for processing the packets read based on the second arbitration scheme.

58. (canceled)

59. (currently amended) The system of claim 1, wherein at least one of the first arbitration element or the second arbitration element is configured to be reprogrammed ~~based on an input regarding a~~ when the speed of at least one of the streams changes.

60. (canceled)

61. (currently amended) The method of claim 21, further comprising:
receiving an input ~~regarding a~~ that the speed of at least one of the variable-speed streams has changed; and

reprogramming at least one of the first arbitration element or the second arbitration element based on the received input.

62. (new) The system of claim 59, wherein the at least one of the first arbitration element or the second arbitration element is reprogrammed to change the number of the first or second entries assigned to the one of the streams.

63. (new) The method of claim 61, wherein reprogramming the at least one of the first arbitration element or the second arbitration element includes changing the rate at which data associated with the one of the streams is stored in or read from the memory.